

PICTURE PROCESSING SYSTEM AND METHOD FOR CONTROLLING THE SAME AND COMPUTER PROGRAM AND STORAGE MEDIUM

Publication number: JP2003101793

Publication date: 2003-04-04

Inventor: NAKAYAMA TADAYOSHI; OTA KENICHI; ITO NAOIKI;
KATO SHINICHI; OSAWA HIDESHI

Applicant: CANON KK

Classification:

- international: H04N7/26; H04N1/413; H04N7/24; H04N7/26;
H04N1/413; H04N7/24; (IPC1-7): H04N1/413; H04N7/24

- European:

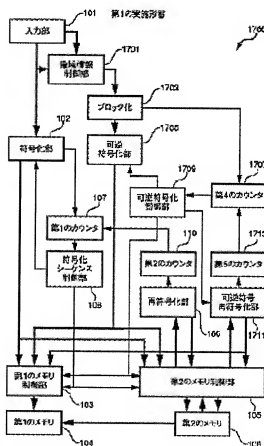
Application number: JP20010285682 20010919

Priority number(s): JP20010285682 20010919

Report a data error here

Abstract of JP2003101793

PROBLEM TO BE SOLVED: To encode multi-level picture data and the image region information in a target size without re-inputting the image region information of multi-level pictures. **SOLUTION:** Multi-level picture data from an input part 101 are encoded by an encoding part 102, and the result is stored in a first memory 104 and a second memory 106. An encoding sequence control part 108 monitors generated code quantity, and at the time of judging that it exceeds a target value, the encoding sequence control part 108 makes the encoding part 102 to continue to encode the picture data to be inputted afterwards by setting a higher compressibility. Also, the data just before it is judged that the code quantity exceeds the target value are decoded and re-encoded with the high compressibility by a re-encoding part 109, and stored in the first memory. At the time of judging that the target value is over, the encoding is continued by converting a part of image region information to be inputted afterwards into a fixed value in order to realize more efficient encoding.



Data supplied from the esp@cenet database - Worldwide

(51) Int.Cl. ⁷	識別番号	F I	テラコト* (参考)
H 0 4 N	1/413	H 0 4 N	5 C 0 5 9
	7/24		Z 5 C 0 7 8

審査請求 未請求 請求項の数 9 O L (全 21 頁)

(21) 出願番号 特願2001-285682(P2001-285682)

(22) 出願日 平成13年9月19日(2001.9.19)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子 3 丁目30番 2 号

(72) 発明者 中山 忠義

東京都大田区下丸子 3 丁目30番 2 号 キヤ
ノン株式会社内

(73) 発明者 太田 健一

東京都大田区下丸子 3 丁目30番 2 号 キヤ
ノン株式会社内

(74) 代理人 100076428

弁理士 大塚 康徳 (外 3 名)

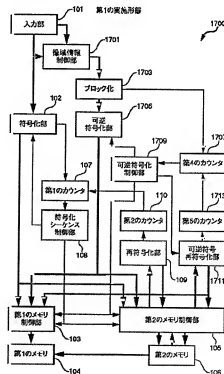
最終頁に続く

(54) 【発明の名称】 画像処理装置及びその制御方法及びコンピュータプログラム及び記憶媒体

(57) 【要約】 (修正有)

【課題】 多値画像の像域情報を再入力し直すことなく、多値画像データ及びその像域情報を目的とするサイズに収まるよう符号化する。

【解決手段】 入力部 101 からの多値画像データは、符号化部 102 で符号化し、その結果を第 1 のメモリ 104、第 2 のメモリ 106 に格納していく。符号化シーケンス制御部 108 は、発生した符号量を監視していて、それが目標値を越えたと判断した場合には、それ以降に入力される画像データに対してはより高圧縮率となるよう符号化部 102 に設定して符号化を続行させる。また、目標値を越えたと判断した直前までのデータについては、再符号化部 109 で復号化し、高い圧縮率で再符号化させ、それを第 1 のメモリに格納する。目標値オーバーしたと判断した場合には、より高い符号を可能とするため、それ以降に入力される像域情報の一部を固定値に変換する処理を行い、符号化を続行する。



【特許請求の範囲】

【請求項1】 多値画像データ及び多値画像データの各画素の像域情報を入力し、それぞれに独立した多値画像圧縮手段と像域情報圧縮手段とを備える画像処理装置であって、

圧縮符号化された多値画像データ及び像域情報を格納するメモリを備え、

前記多値画像圧縮手段は、

圧縮率を決定するパラメータが変更可能な第1の多値画像圧縮手段と、

圧縮率を決定するパラメータが変更可能であって、前記第1の多値画像圧縮手段で圧縮した符号データを復号し、再圧縮する第2の多値画像圧縮手段と、

前記第1の多値画像圧縮手段によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する画像データ符号量監視手段と、

該画像データ符号量監視手段によって前記所定量に達したと判断した場合、前記第1、第2の多値画像圧縮手段へ圧縮率を高くするパラメータを設定する画像符号パラメータ設定手段と、

該画像符号パラメータ設定手段によりパラメータを変更した場合、前記第1の多値画像圧縮手段で従前に生成された符号データを前記第2の多値画像圧縮手段によって再符号化させ、当該再符号化を済ませた符号データを、前記第1の多値画像圧縮手段のパラメータ変更後の符号データとして前記第1のメモリに保存させると共に、パラメータ変更後の前記第1の多値画像圧縮手段で生成された符号化データを、後続符号データとして前記メモリに保存させる画像圧縮制御手段とを備え、

前記像域情報圧縮手段は、

前記像域情報を可逆符号化する第1の像域情報符号化手段と、

該第1の像域情報符号化手段で符号化された像域情報を再度可逆符号化する第2の像域情報符号化手段と、

前記第1の像域情報圧縮手段によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する像域情報量監視手段と、

該像域情報量監視手段によって前記所定量に達したと判断した場合、前記第1、第2の像域情報圧縮手段へ圧縮率を高くするパラメータを設定する像域符号パラメータ設定手段と、

該像域符号パラメータ設定手段によりパラメータを変更した場合、前記第1の像域情報圧縮手段で従前に生成された符号データを前記第2の像域情報圧縮手段によって再符号化させ、当該再符号化を済ませた符号データを、前記第1の像域情報圧縮手段のパラメータ変更後の符号データとして保存させると共に、

パラメータ変更後の前記第1の像域情報圧縮手段で生成された符号化データを、後続符号データとして前記メモリに保存させる像域圧縮制御手段とを備えることを特徴

とする画像処理装置。

【請求項2】 前記画像圧縮制御手段は、目標データ量を越えたと判断する度に、符号化する際の量子化ステップを従前よりも大きく設定することを特徴とする請求項第1項に記載の画像処理装置。

【請求項3】 前記像域圧縮制御手段は、目標データ量を越えたと判断する度に、像域情報を構成する像域成分データの一部を固定値に置き換えてエントロピーを減少させることを特徴とする請求項第1項に記載の画像処理装置。

【請求項4】 前記像域成分データには、注目画素が文字・線画領域であるか中間調領域にあるかを示すデータ、注目画素が有彩色であるか無彩色であることを示すデータが含まれることを特徴とする請求項第3項に記載の画像処理装置。

【請求項5】 前記像域符号量監視手段が、符号データ量が目標データ量を越えたと最初に判断した場合、前記像域圧縮制御手段は中間調領域にあって無彩色であることを示す画素の像域成分情報を有彩色に変更するよう要求し、

前記像域符号量監視手段が符号データ量が目標データ量を越えたと2回目に判断した場合、前記像域圧縮制御手段は文字線画にあって無彩色であることを示す画素の属性情報を有彩色に変更するよう要求することを特徴とする請求項第4項に記載の画像処理装置。

【請求項6】 更に、多値画像データを入力し、当該入力した多値画像データの解析することで、各画素毎に、文字・線画領域であるか中間調領域かを示す属性成分データ、及び、有彩色／無彩色のいずれであるかを示す属性成分データを前記属性情報として出力する属性情報生成手段を備えることを特徴とする請求項第4項に記載の画像処理装置。

【請求項7】 多値画像データ及び多値画像データの各画素の像域情報を入力し、それぞれに独立した多値画像圧縮工程と像域情報圧縮工程とを備える画像処理装置の制御方法であって、

前記多値画像圧縮工程は、

圧縮率を決定するパラメータが変更可能な第1の多値画像圧縮工程と、

圧縮率を決定するパラメータが変更可能であって、前記第1の多値画像圧縮工程で圧縮した符号データを復号し、再圧縮する第2の多値画像圧縮工程と、

前記第1の多値画像圧縮工程によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する画像データ符号量監視工程と、

該画像データ符号量監視工程によって前記所定量に達したと判断した場合、前記第1、第2の多値画像圧縮工程に圧縮率を高くするパラメータを設定する画像符号パラメータ設定工程と、

該画像符号パラメータ設定工程によりパラメータを変更

した場合、
前記第1の多値画像圧縮工程で従前に生成された符号データを前記第2の多値画像圧縮工程によって再符号化させ、当該再符号化を済ませた符号データを、前記第1の多値画像圧縮工程のパラメータ変更後の符号データとして所定のメモリに保存させると共に、
パラメータ変更後の前記第1の多値画像圧縮工程で生成された符号化データを、後続符号データとして前記メモリに保存させる画像圧縮制御工程とを備え、
前記領域情報圧縮工程は、
前記領域情報を可逆符号化する第1の領域情報符号化工程と、
該第1の領域情報符号化工程で符号化された領域情報を再度可逆符号化する第2の領域情報符号化工程と、
前記第1の領域情報圧縮工程によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する領域情報量監視工程と、
該領域情報量監視工程によって前記所定量に達したと判断した場合、前記第1、第2の領域圧縮工程に圧縮率を高くするパラメータを設定する領域符号パラメータ設定工程と、
該領域符号パラメータ設定工程によりパラメータを変更した場合、前記第1の領域情報圧縮工程で従前に生成された符号データを前記第2の領域情報圧縮工程によって再符号化させ、当該再符号化を済ませた符号データを、前記第1の領域情報圧縮手段のパラメータ変更後の符号データとして前記メモリに保存させると共に、
パラメータ変更後の前記第1の領域情報圧縮工程で生成された符号化データを、後続符号データとして前記メモリに保存させる領域圧縮制御工程と備えることを特徴とする画像処理装置の制御方法。
【請求項8】 多値画像データ及び多値画像データの各画素の領域情報を入力し、それぞれを独立して圧縮させる多値画像圧縮工程のプログラムと領域情報圧縮工程のプログラムで構成されるコンピュータプログラムであって、
前記多値画像圧縮工程のプログラムは、
圧縮率を決定するパラメータが変更可能な第1の多値画像圧縮工程のプログラムと、
圧縮率を決定するパラメータが変更可能であって、前記第1の多値画像圧縮工程で圧縮した符号データを復号し、再圧縮する第2の多値画像圧縮工程のプログラムと、
前記第1の多値画像圧縮工程によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する画像データ符号量監視工程のプログラムと、
該画像データ符号量監視工程によって前記所定量に達したと判断した場合、前記第1、第2の多値画像圧縮工程に圧縮率を高くするパラメータを設定する画像符号パラ

メータ設定工程のプログラムと、
該画像符号パラメータ設定工程によりパラメータを変更した場合、前記第1の多値画像圧縮工程で従前に生成された符号データを前記第2の多値画像圧縮工程によって再符号化させ、当該再符号化を済ませた符号データを、前記第1の多値画像圧縮工程のパラメータ変更後の符号データとして所定のメモリに保存させると共に、
パラメータ変更後の前記第1の多値画像圧縮工程で生成された符号化データを、後続符号データとして前記メモリに保存させる画像圧縮制御工程のプログラムとを備え、
前記領域情報圧縮工程のプログラムは、
前記領域情報を可逆符号化する第1の領域情報符号化工程のプログラムと、
該第1の領域情報符号化工程で符号化された領域情報を再度可逆符号化する第2の領域情報符号化工程のプログラムと、
前記第1の領域情報圧縮工程によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する領域情報量監視工程のプログラムと、
該領域情報量監視工程によって前記所定量に達したと判断した場合、前記第1、第2の領域圧縮工程に圧縮率を高くするパラメータを設定する領域符号パラメータ設定工程のプログラムと、
該領域符号パラメータ設定工程によりパラメータを変更した場合、前記第1の領域情報圧縮工程で従前に生成された符号データを前記第2の領域情報圧縮工程によって再符号化させ、当該再符号化を済ませた符号データを、前記第1の領域情報圧縮手段のパラメータ変更後の符号データとして前記メモリに保存させると共に、
パラメータ変更後の前記第1の領域情報圧縮工程で生成された符号化データを、後続符号データとして前記メモリに保存させる領域圧縮制御工程のプログラムと備えることを特徴とするコンピュータプログラム。
【請求項9】 請求項第8項に記載のコンピュータプログラムを格納することを特徴とするコンピュータ可読記憶媒体。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】本発明は、画像データを圧縮符号化する画像処理装置及びその制御方法及びコンピュータ、並びにコンピュータ可読記憶媒体に関するものである。
【0002】
【従来の技術】従来、静止画像の圧縮方式には、離散コサイン変換を利用したJPEG方式や、Wavelet変換を利用した方式が多く使われている。この種の符号化方式は、可変長符号化方式であるので、符号化対象の画像毎に符号量が変化する。
【0003】国際標準化方式であるJPEG方式では、

画像に対して1組の量子化マトリクスしか定義できないので、プリスキャン無しには、符号量調整が行えず、限られたメモリに記憶するシステムで使用する場合には、メモリオーバーを起こす危険性があった。

【0004】これを防止するために、予定した符号量よりオーバーした場合は、圧縮率を変更して、原稿の再読み込みを行う方法や、予めプリスキャンによる符号量見積もりを行ない、符号量を調整するために、量子化パラメータの再設定を行う方法などがとられていた。

【0005】一方、画像情報には、元の画像データ以外に、該画像データに付随する像域情報というものがある。像域情報は主に、画像出力時の見栄えを良くするために、画像出力部で色処理や諧調数の調整に用いられる。有彩色と無彩色が混在する自然画像と、原稿中に多く見られる黒文字とでは、同じ黒色でも使用するインクの種類を変えることで、自然画像を自然画像らしく見せる一方で、鮮明な文字を出力することが出来る。

【0006】このように、画素毎に、有彩色か無彩色、文字部かそうでないか、といった各々1ビットの属性フラグデータを持つことにより、画像出力時、特にプリントアウト時に出力画像の画質向上を図ることが出来る。像域情報には前記以外の他の情報もある。

【0007】画像情報を圧縮するには、画像データの圧縮はもちろんのこと、上記像域情報も圧縮する必要がある。像域情報は2値データの集まりであり、これを圧縮するには基本的に可逆の符号化方式を用いる必要がある。従来、像域情報の圧縮には Packbits や JBIG 符号化方式が用いられていた。

【0008】しかしながら、前記像域情報をこれらの符号化方式で圧縮するだけでは、符号量調整を行わず、限られたメモリに記憶するシステムで使用する場合には、メモリオーバーを起こす危険性があり、大きな問題であった。

【0009】【発明が解決しようとする課題】しかしながら、従来は、画像データの圧縮ばかりが検討され、像域情報の圧縮については、検討されることが少なかった。まして、該像域情報の圧縮後の符号量を目値以内に取りめること等、ほとんど考えられることが無く、ある符号化方式を用いて該像域情報を単純に符号化するだけであった。

【0010】本発明は上記従来例に鑑みて成されたものであり、画像データ及び像域情報それぞれを、それぞれの日値以内の符号量に取りめるための効果的な画像処理装置及びその制御方法及びコンピュータプログラム及びコンピュータ可読記憶媒体を提供しようとするものである。

【0011】

【課題を解決するための手段】かかる課題を解決するための、例えば本発明の画像処理装置は以下の構成を備える。すなわち、多値画像データ及び多値画像データの各

画素の像域情報を入力し、それぞれが独立した多値画像圧縮手段と像域情報圧縮手段とを備える画像処理装置であって、圧縮符号化された多値画像データ及び像域情報を格納するメモリを備え、前記多値画像圧縮手段は、圧縮率を決定するパラメータが変更可能な第1の多値画像圧縮手段と、圧縮率を決定するパラメータが変更可能であって、前記第1の多値画像圧縮手段で圧縮した符号データを復号し、再圧縮する第2の多値画像圧縮手段と、前記第1の多値画像圧縮手段によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する画像データ符号量監視手段と、該画像データ符号量監視手段によって前記所定量に達したと判断した場合、前記第1、第2の多値画像圧縮手段に圧縮率を高くするパラメータを設定する画像符号パラメータ設定手段と、該画像符号パラメータ設定手段によりパラメータを変更した場合、前記第2の多値画像圧縮手段によって前記第1の多値画像圧縮手段で従前に生成された符号データを再符号化させ、当該再符号化後の符号データを、前記第1の多値画像圧縮手段のパラメータ変更後の符号データとして前記第1のメモリに保存させると共に、パラメータ変更後の前記第1の多値画像圧縮手段で生成された符号化データを、後続符号データとして前記メモリに保存させる画像圧縮制御手段とを備え、前記像域情報圧縮手段は、前記像域情報を可逆符号化する第1の像域情報符号化手段と、該第1の像域情報符号化手段で符号化された像域情報を再度可逆符号化する第2の像域情報符号化手段と、前記第1の像域情報圧縮手段によって生成される符号量を監視すると共に、当該符号データ量が所定量になったか否かを判断する像域情報量監視手段と、該像域情報量監視手段によって前記所定量に達したと判断した場合、前記第1、第2の像域圧縮手段に圧縮率を高くするパラメータを設定する像域符号パラメータ設定手段と、該像域符号パラメータ設定手段によりパラメータを変更した場合、前記第2の像域情報圧縮手段によって前記第1の像域情報圧縮手段で従前に生成された符号データを再符号化させ、当該再符号化後の符号データを、前記第1の像域情報圧縮手段のパラメータ変更後の符号データとして保存させると共に、パラメータ変更後の前記第1の像域情報圧縮手段で従前に生成された符号化データを、後続符号データとして前記メモリに保存させる画像圧縮制御手段とを備えることを特徴とする。

【0012】

【発明の実施の形態】以下、添付図面に従って本発明に係る実施形態を説明するが、先ず、基本部分について説明する。

【0013】図1は、実施形態が適用する画像処理装置100の機能ブロック構成図である。以下、同図の各部を簡単に説明する。

【0014】画像処理装置100は、イメージスキャナから画像を入力する入力部101を備えている。なお、

入力部101は、ページ記述言語レンダリングなどから画像データを入力しても良いし、記憶媒体に格納された画像ファイルを読込むことで実現しても良く、場合によってはネットワークより受信するようにしても良い。

【0015】符号化部102は、入力された画像データの符号化を行なう。なお、符号化方式は公知のJPEG符号化方式を用い、8×8画素単位に相当する画像データを直交変換し、後述する量子化ステップを用いた量子化、ハフマン符号化処理を行なうものである。

【0016】第1のメモリ制御部103と第2のメモリ制御部105は、上記符号化部102から夫々に出力されてくる上記符号化データ（同じ符号化データ）を第1のメモリ104と第2のメモリ106へ格納する様に制御する。ここで、第1のメモリ104は、最終的に確定した（目標値以内のデータ量に圧縮し終わった）符号化データを、図1の基本構成の外部に接続されるネットワーク機器、画像出力装置や大容量記憶装置等へ出力するために、該符号化データを保持するためのメモリである。また、第2のメモリ106は、前記符号化データを第1のメモリ上に形成するための圧縮符号化処理を補助する作業用のメモリである。

【0017】カウンタ107は、符号化部102によって圧縮符号化された画像データのデータ量をカウントし、該カウント値を保持すると共に、そのカウント結果を符号化シーケンス制御部108に出力する。

【0018】符号化シーケンス制御部108では、カウンタ107のカウント値がある設定値に達したかどうかを検出し、その設定値に達した（目標値を越えた）ことを検出した時にメモリ104内の格納済みのデータを廃棄するよう第1のメモリ制御部103に制御信号を出力する。上記第1のメモリ制御部103は、この制御信号に基づいて、メモリアドレスカウンタをクリアするか、あるいは符号化データ管理テーブルをクリアすることにより、前記格納データを廃棄する。また、このとき、符号化シーケンス制御部108は、第1のカウント107をゼロクリアする（入力部101からの入力は継続している）と共に、符号化部102に対して今までより、高い圧縮率で符号化を行なうよう制御する。すなわち、本装置の符号化処理で発生する符号化データのデータ量が最終的に例えば1/2になるように制御する。なお、ここでは、1/2としたが任意に設定できることは言うまでもない。

【0019】そして、圧縮率変更後の符号化データも、これまでも同様、第1のメモリ制御部103と第2のメモリ制御部105を経て、第1のメモリ104と第2のメモリ106に夫々格納される。

【0020】さらに、符号化シーケンス制御部108は、第2のメモリ制御部105に対して、これまでに第2のメモリ106に格納した符号化データを読み出し、

符号化データ変換手段である再符号化部109に該符号化データを出力するよう制御信号を出す。

【0021】再符号化部109は、入力された符号化データを復号化し、データ量を減らすための再量子化等を行なった後に再び符号化処理を行ない、圧縮率が変更された符号化データと同じ圧縮率のデータ量を第2のカウント110に出力する。

【0022】この再符号化部109から出力される符号化データは、第1のメモリ制御部103と第2のメモリ制御部105を経由して、それぞれ、第1のメモリ104と第2のメモリ106に格納される。

【0023】再符号化処理が終了したかどうかは、第2のメモリ制御部が検出する。すなわち、再符号化処理するために読み出すデータが無くなれば、再符号化処理の終了を符号化シーケンス制御部108に知らせる。実際には、第2のメモリ制御部105の読みだし処理だけでなく、再符号化部109の処理も終了した後に、符号化処理が完了したことになる。

【0024】第2のカウント110で得られるカウント値は、再符号化処理が完了した後、第1のカウント107で保持されているカウント値に加算される。この加算結果は再符号化処理が完了した直後における、第1のメモリ104内のデータ量の合計を表す。即ち、1画面分の符号化部102と再符号化部109の符号化処理が終了した時点では、上記加算後の第1のカウント107で保持されているカウント値は、1画面分を本装置が符号化した場合に発生した総データ量を表す（詳細は後述）。

【0025】符号化部102は、再符号化処理の終了/未終了に関わらず、符号化するべき入力部101からの画像データが残っている限りは符号化処理を継続して行なう。

【0026】カウンタ107のカウント値がある設定値に達したかどうかは入力部101から入力された1ページの画像データの符号化処理（符号化、再符号化）が終わるまで繰り返され、上述した符号化と再符号化の処理は、ここで得られる検出結果に応じた制御の上で実行される。

【0027】上記、図1の構成における処理のフローを表わすフローチャートを図8に示すが、説明を簡単にするため、簡略化した図3のフローチャートに従って先ず説明する。

【0028】既に説明したように、本発明の画像処理装置100は、スキャナ等の入力部101から入力した1ページの画像データを所定のデータ量以下に圧縮符号化する装置である。該符号化処理を実現するために、前記入力部101以外に、符号化部102、再符号化部109、第1のメモリ104、第2のメモリ106等有する。これらの機能ブロックを用い、図3に示すフローチャートに基づいて符号化処理を行なう。

【0029】図3のフローチャートは、大別すると、下記の3つの処理フェーズに分かれる。

- (1) 符号化フェーズ
- (2) 符号化・再符号化フェーズ
- (3) 転送フェーズ

上記それぞれ処理フェーズにおいて、どのように画像データ、符号化データ等が流れて処理され、メモリにどのように格納されるかを視覚的に解り易く示したのが図4乃至図7である。

【0030】図4は、図3のフローチャートにおけるステップS303とS305に対応する符号化フェーズの初期状態を表わす。また、図5はステップS307～S315に対応する符号化・再符号化フェーズの処理状態を、図6はステップS317に対応する転送フェーズの処理状態を、図7は転送フェーズ後の符号化フェーズの処理状態を表わす。以下、各フェーズについて説明する。

【0031】＜＜符号化フェーズ＞＞1ページ分の画像データの符号化処理は、符号化パラメータの初期設定（ステップS301）から始まる。ここでは符号化処理する画像サイズ（スキャナ等の入力部101から読み取る用紙サイズ）から一意的に定まる符号化データ量の上限值や符号化部102（ここでは公知のJPEG符号化方式を用いるものとする）に適用する量子化ステップ（Q1）といったパラメータを設定する。

【0032】そして、ステップS303にて、第1のカウント107は、実際の符号化処理（画像の8×8画素単位にJPEG圧縮）を行ない、出力される符号化データのデータ量を累積カウントする。

【0033】次にステップS305にて、該データ量のカウント値が上記上限値をオーバーしたかどうかを検知し、オーバーしていなければステップS303のJPEG符号化処理を継続する。これが初期状態の符号化フェーズである。

【0034】符号化部102から出力する符号化データは、図4に示すように第1のメモリ104と第2のメモリ106の両方に格納されていく。縦縞で示した領域が該格納した符号を表現している。

【0035】＜＜符号化・再符号化フェーズ＞＞符号化部102の符号化処理が進行し、前記データ量のカウント値が設定されている上限値をオーバーすると、ステップS307にて、第1のメモリ104内の符号化データを廃棄すると共に、ステップS309にて、符号化部102の量子化ステップをQ2に変更する。

【0036】符号化データのデータ量のカウント値が設定された上限値をオーバーするという事は、圧縮後のデータ量が目標値以内に取まらないことを意味する。よって同じ量子化ステップを用いて符号化処理を継続しても意味が無いので、前よりもデータ量が少なくなるように、Q1よりも量子化ステップ幅の大きい量子化ステップ

Q2に変更するわけである。

【0037】量子化ステップを変更した後、ステップS311では符号化部102の符号化処理を再開し、図5に示すように符号化データを第2のメモリ106のみに格納する。それと並行して、ステップS313の再符号化処理を行なう。再符号化処理では、第2のメモリ106に格納済みの符号化データを読み出して、再符号化部109にて再符号化処理を行ない、前記2つのメモリ104、106に格納する。そして、縦縞の符号を全て再符号化するまで、該符号化処理と再符号化処理を継続する。再符号化部109から出力される再符号化データは、量子化ステップ変更後に符号化部102から出力される符号化データと同じ量子化ステップで符号化して得られる符号化データと全く同一の符号化データである。

【0038】具体的にこの再符号化処理では、符号化データを一旦ハフマン復号した後の各量子化値に対して、これら値を2で割った結果と同様の結果が出るビットシフト処理を施した後、再度ハフマン符号化を行なうことにより実現される。この方法は、ビットシフトのみで量子化ステップを変更する点と逆置交変換や再置交変換処理を行なわない点で、高速な再符号化処理が可能である。ステップS315では、再符号化処理の終了検知が行なわれる。

【0039】再符号化後のデータ量は再符号化前の符号化データのデータ量よりも少なくなるので、図5に示すように、再符号化前の符号を格納していたメモリ領域に再符号化後の符号化データを上書きするように格納することができる。再符号化処理が終了した時点で、縦縞の符号化データのデータ量は図6に示すの斜め縞の符号化データのデータ量へと減少する。

【0040】以上で説明したステップS307～S315が、符号化・再符号化フェーズで行なう処理である。

【0041】＜＜転送フェーズ＞＞再符号化処理が終了したら、ステップS317では転送処理が行なわれる。該転送処理では、図6に示すように、符号化・再符号化フェーズで第2のメモリ106のみに格納した斜め縞の符号化データを、第1のメモリ104内の斜め縞の符号化データに連結されるアドレスに転送し、格納する。その一方で、第2のメモリ106上で分散してしまっている斜め縞の符号化データと斜め縞の符号化データが第1のメモリ104上で連続して格納される様に、前記斜め縞の符号化データを第2のメモリ106内で転送し、連結させる。これが、転送フェーズで行なう処理である。

【0042】上記転送フェーズが終了したら、ステップS303、S305の符号化フェーズに戻り、図7に示すように斜め縞の符号を符号化部102から出力して2つのメモリ104、106に格納する。この符号化フェーズは、初期状態の符号化フェーズ（図4）と少し異なり、符号化部102で符号化する際の量子化ステップ

がQ1からQ2に変更されていると共に、2つのメモリ104、106に格納されている符号化データも様々なフェーズで処理された符号の集まりである。それらの違いを無視すれば、転送フェーズ直後の符号化フェーズと初期状態の符号化フェーズは、同じと見なせる。

【0043】よって、符号化フェーズ、符号化・再符号化フェーズと転送フェーズの3つを繰り返すことで、最終的に1ページの画像データをデータ量設定値以下に圧縮した符号を第1のメモリに格納することが出来る。しかも、入力部101は一連の処理が終わるまで、入力を継続するだけである。すなわち、画像を再度最初から入力し直すということが無くなる。

【0044】図3に示したフローチャートは、説明が理解しやすいように、図4、図5、及び、図6に示した各フェーズに対応する処理のみを記述した。しかしながら実際には、1ページの画像データの入力はどこかのフェーズで終了する。従って、どのフェーズで終了したかによって、それ以降の対応も多少異なる。それを考慮した流れを示したのが図8のフローチャートである。図8のフローチャートは、1ページ分の画像データの入力完了と図3で説明した各種処理との関係を検討したものであり、ここでは図3のフローチャートに、ステップS801、S803、S805、S807を追加している。

【0045】ステップS801、S803、S805は、それぞれ、符号化フェーズ、符号化・再符号化フェーズ、転送フェーズにおいて、入力部101からの1ページの画像データの入力が終了したことを検知する。

【0046】符号化フェーズと転送フェーズで1ページ分の画像データの入力が終了したことを検知した場合（ステップS801、S805）、ステップS807へ移り、当該ページの圧縮符号化処理を終了し、次に処理すべき1ページ以上の画像データがあれば、次の1ページの画像データの圧縮符号化処理を開始し、無ければ停止状態に入る。

【0047】一方、符号化・再符号化フェーズで1ページの画像データの入力終了を検知した場合（ステップS803）には、符号化部102では再符号化処理する画像データが無くなるまで一旦動作を止める必要があるため、ステップS311の符号化処理をパスし、ステップS313で、今までに符号化部102で符号化済みの画像データを所定の符号化データ量に抑える為の再符号化処理のみを継続して行なう。再符号化処理が全て終了して、その後の転送処理が終わらないうと、1ページ分の画像データ全体の符号化データが第1のメモリ上に集まらないため、1ページ分の画像データの入力終了後も再符号化処理及びそれに続く転送処理は継続して行われる必要がある。この場合には、ステップS315にて、再符号化処理が全て終了したことを検知すると、符号化・再符号化フェーズ中に、第2のメモリ106のみに格納された符号化データを第1のメモリに転送し（ステップ

S317）た後、次のステップS805にて、1ページ分の画像データの入力終了を検知されてステップS807へ移ることになる。

【0048】以上が動作であり、図8の動作説明でもある。

【0049】＜メモリ格納方法の変形例＞図9、図10は図5、図6の概念図で示したメモリ格納方法の変形例を示す図である。

【0050】図5の概念図においては、符号化・再符号化フェーズでは、符号化部102から出力する符号化データは第2のメモリ106のみに格納していたが、図9に示すように符号化・再符号化フェーズ中に、符号化部102から出力する符号化データを第1、第2メモリの両方に直接格納する。

【0051】符号化部102から見ると、どのフェーズで符号化して出力する符号化データも両方のメモリへ格納することになる。また、図6の概念図とは異なり、図10に示す様に、転送フェーズでメモリ間のデータ転送が必要なくなる。またこの変形例の場合には、符号化・再符号化フェーズにおいて、符号化データと再符号化データを第1のメモリ104へ送った順序で順次格納される。そのため2種類のデータが入り混じってしまうという問題は有る。

【0052】従って、この変形例の場合にはこれに対応する為に符号化データをある単位で区切って、ファイル或いはバケットとして管理する様にする。具体的には、ファイル管理テーブル、或いは、バケット管理テーブル等を別で作成して管理する。

【0053】一つの手法としては、符号化部102からのデータを第1メモリ104に格納する際、適当な単位（例えば前記直交変換の単位が 8×8 のブロックであるので、 $8 \times i$ （ $i=1, 2, \dots$ の整数）ライン分のデータ）毎に、画像データの先頭から管理番号を割り当て、各管理番号に対応する符号化データの格納先頭アドレスと該符号化データ量とを、管理番号順に格納できるように管理テーブルを作成する。

【0054】符号化部102や再符号化部109は処理中のデータの管理番号を保持し、該管理番号に基づいて、符号化データ格納時の先頭アドレスと符号化データ量とを管理テーブルに書き込む。このようにすれば、符号化部102と再符号化部109で処理した符号化データをランダムに格納したとしても、前記管理テーブルを管理番号順にアクセスし、その時読み出させる先頭アドレスと符号化データ量に基づいて、符号化データ第1メモリ104から読み出せば、画像の先頭から順番に符号化データを読み出すことができる。このような管理機構を設ければ、画像上で連続するデータをメモリ上で連続するように格納する必要性が無くなる。

【0055】図10の概念図における転送フェーズ後の符号化フェーズは、これまで説明した2つの符号化フェ

ーズ(図4、図7)とほとんど同じであり、第1のメモリ内における符号の格納状態が図11に示した様に若干異なるだけである。よって、先の説明と本変形例は、3つのフェーズを繰り返して処理することに変わりはない。

【0056】次に、本発明において特徴的な符号化処理を行なう為の、第2の基本構成の例(これまで説明した構成を第1の例という)を図2を用いて説明する。

【0057】図2は、第2の例における画像処理装置200のブロック構成図である。

【0058】図1の画像処理装置100と大きく異なる点は、最初に符号化を行なう符号化部が2つ並列に存在する点である。画像処理装置200は、入力部201から入力される画像データを、第1の符号化部202と第2の符号化部205で並行して符号化し、互いに圧縮率の異なる2種類の符号化データを生成する。本例でも、符号化方式は公知のJPEG符号化方式を用い、8×8画素単位に相当する画像データを直交変換し、後述する量子化ステップを用いた量子化、ハフマン符号化処理を行なうものである。

【0059】なお、本例では第1の符号化部202よりも、第2の符号化部205の方が適用する圧縮率を高く設定する場合について説明する。具体的には、第1の符号化部202における量子化ステップをQ1、第2の符号化部205の量子化ステップをQ2(=2×Q1)とする。

【0060】符号化部202から出力される符号化データは、第1のメモリ制御部203を経由して、第1のメモリ204に格納される。このとき、第1のカウント208は、符号化部202から出力される符号化データのデータ量をカウントし、これを保持すると共に、符号化シーケンス制御部209にも出力する。

【0061】一方、符号化部205で符号化された符号化データは、第2のメモリ制御部206を経由して、第2のメモリ207に格納される。このとき、第2のカウント210は、符号化部205から出力される符号化データのデータ量をカウントし、これを保持する。更に、後述する第2のメモリ207に格納している符号化データを第1のメモリ204に転送する時には、それと同時に上記カウント値を、第1のカウント208に転送する。

【0062】さて、第1のカウント208が符号化部202から出力される符号化データのデータ量をカウント中に、該カウント値がある設定値に達した時には、符号化シーケンス制御部209は、第1の例と同様、メモリ制御部203に対してメモリ204に格納されているデータを廃棄するよう制御信号を出す。

【0063】そして、符号化シーケンス制御部209は、第2のメモリ207に格納している符号化データを読み出して第1のメモリ204に転送し、第1のメモリ

204に格納するよう、メモリ制御部206とメモリ制御部203に制御信号を出力する。この結果、第2のカウント210のカウント値が第1のカウント208に転送され、その値が第1のカウントのカウント値としてロード(上書き)される。

【0064】要するに、上記第2のカウント210のカウント値は、第2のメモリ207に格納している符号化データのデータ量を表わしているため、そのカウント値と符号化データを、互いの対応付けが変わらない様に、そのまま第1のカウントと第1のメモリへコピーしたと考えれば良い。

【0065】さらに、符号化シーケンス制御209は、第1の符号化部202および、第2の符号化部205に対して、今までよりも、符号化データが少なくなるような符号化を行なうように制御信号を出す。

【0066】例えば、第1の符号化部202、及び、第2の符号化部205における量子化ステップQを2倍に切り替える。この結果、第1の符号化部202は、その直前までの第2の符号化部205における量子化ステップQ2(=2×Q1)を継承することになり、第2の符号化部205は更に大きな量子化ステップQ2×2を用いて、次のオーバーフローに備えた更に高い圧縮率の符号化処理を行うことになる。

【0067】ここでは、量子化ステップの倍率比を2倍としたがこれに限らず、任意に設定できることは示すまでもない。切り替えられた各符号化部202、205から出力された符号化データは、それぞれ、対応するメモリ制御部203、206を経由して、対応するメモリ204、207に格納される。

【0068】そして、符号化シーケンス制御209は、メモリ制御部206に対し、既に第2のメモリに格納している符号化データを読み出して、再符号化部211にデータを送るよう制御信号を出す。再符号化部211は、図1の再符号化部109と同様にして符号化データの再符号化処理を行なう。

【0069】第3のカウント212は、再符号化部211が出力したデータ量をカウントするもので、再符号化処理を開始する直前にゼロにリセットされ、再符号化処理中の出力データ量をカウントする。このカウント212は、再符号化処理が終了した時点で、そこで得られたカウント値を第2のカウント210に転送する。

【0070】第2のカウント210は、上記転送されてきたデータ量カウント値を、第2のカウント210内に保持しているカウント値に加算することにより、再符号化処理中にメモリ207に格納した、符号化データと再符号化データの合計のデータ量を算出する。即ち、メモリ207に格納しているデータ量とカウント210のカウント値とが一致する。

【0071】再符号化処理の終了/未終了に関わらず、符号化するべき入力部201からの画像データが残って

いれば、2つの符号化部202と205による符号化処理を継続して行なう。そして、カウンタ208のカウンタ値がある設定値に達したかどうかの監視は入力部201から入力される1ページ分の画像データの符号化処理(符号化、再符号化)が終わるまで繰り返され、上述した符号化と再符号化の処理は、ここで得られる検出結果に応じた制御の上で実行される。

【0072】上記図2の構成における処理のフローを表すフローチャートを図12に示す。

【0073】図2で説明したように符号化部が2つある場合は、図12に示すフローチャートに基づいて1ページ分の画像データの符号化を行なう。なお、図12の説明は、符号化部が1つの場合のフローチャートである図8とは、大半は類似しており、当業者であれば上記説明から本第2の例の特徴は十分に理解できるであろうから、符号化部1つの場合と同じように3つのフェーズで処理を説明する様にし、図8と異なる点を主に説明することとする。

【0074】上述した図8のフローと本例のフローとの一番大きな違いは、ステップS317の転送処理が、ステップS317とステップS309の間に移動していることである。要するに、符号化・再符号化フェーズと転送フェーズが入れ替わったと見なせば良い(ステップS307の符号化データの廃棄処理は例外である)。

【0075】ステップS301の符号化パラメータの初期設定では、第1の符号化部202に量子化ステップQ1を、第2の符号化部205には量子化ステップQ2(=2×Q1)を設定する。

【0076】符号化フェーズでは、ステップS801、S303、S305を繰り返して実行する。ステップS801とステップS305は符号化部が1つの場合と同じ処理であるが、ステップS303の符号化処理だけは図13に示すように異なっている。

【0077】第1のメモリ204へ格納する符号化データは圧縮率が段階的に高くなるようにするため、最初に格納する符号化データは圧縮率が一番低い量子化ステップQ1で符号化したデータを格納し、第2のメモリへ格納する符号化データは量子化ステップQ2で符号化したデータを格納する。

【0078】第1のメモリ204へ格納中のデータ量が設定されている上限値をオーバーしたら(ステップS305)、直ちに、第1のメモリ204で保持していた符号化データを廃棄し(ステップS307)、第2のメモリ207で保持している圧縮率の高い符号化データを、第1のメモリ204へ転送する(ステップS317、図14参照)。これにより、第1の例(図1)で説明した1回目の再符号化処理の終了を待たずに、速やかに、上限値をオーバーしない適切な2番目の候補の符号化データを第1のメモリ207内に格納出来る。これが、図1に対する、2つの符号器を持つ図2を適用することの最

大の利点である。

【0079】本第2の例では、2つのメモリ204、207で同じ圧縮率の符号化データを持っていることが無駄という考え方なので、第2のメモリ207には、第1のメモリ204に格納する符号化データよりも圧縮率の高い符号化データを格納しておくようにしている。従って、それ以降の処理もこの考え方に基き行われるものであり、第2のメモリ207内の符号化データを第1のメモリ204に転送する処理(転送フェーズ)が終了した後は、第2のメモリ207の符号化データを、更に1段階圧縮率の高い符号化データを保持する様に再符号化することとなる。

【0080】具体的に、はまず図15に示す様に、転送フェーズの次の符号化・再符号化フェーズでは、上記再符号化の前に、2つの符号化部202、205に適用される各量子化ステップQ1、Q2をそれぞれQ2、Q3へ変更し(ステップS309)、1ページの画像データの入力が終了せずに続いていれば(ステップS803)、後続の画像データは新たな量子化ステップが設定された2つの符号化部で該入力データを符号化して(ステップS311)、対応する各メモリ204、207へ格納する。そして、上記符号化処理と並行して第2のメモリに格納されている符号化データ(第1のメモリ204に転送したものは、第1のメモリ内の符号化データよりも1段階高い圧縮率の符号化データに変更するべく、再符号化部211にて量子化ステップQ3を用いて符号化されたデータが得られる様な再符号化処理(S313)を行ない、再符号化データを第2のメモリ207に格納し直す。

【0081】なお、本第2の例でも、第1の例と同様、再符号化処理では、符号化データを一旦ハフマン復号した後の各量子化値に対して、これら値を2で割った結果と同様の結果が出るビットシフト処理を施した後、再度ハフマン符号化を行なうことにより実現される。この方法は、ビットシフトのみで量子化ステップを変更する点と逆直交変換や再直交変換処理を行わない点で、高速な再符号化処理が可能である。

【0082】なお、本第2の例の様に符号化部が2つある場合には、図15に示したように、第2のメモリ207に符号化データと再符号化データを混在して格納する状況が発生する。従って、前述したように、符号化データはある単位で区切って、ファイル或いはバケットとして管理することが、第2のメモリ207に対して必要になる。その為には、例えば第1の例における変形例と同様の構成を設ければ良いであろう。

【0083】図12において、再符号化処理の終了をステップS315で検知したら、また符号化フェーズ(ステップS801、S303)に移行する。なお、符号化・再符号化フェーズ後の符号化フェーズでは、図16に示すように、2つのメモリ204、207が保持する符

号化データは圧縮率が違うだけでなく、符号化データの混在の仕方（アドレス）もかなり違ってくる。従って、再度、第1のメモリ204のデータ量が設定値をオーバーした場合には、第2のメモリ207で保持されている符号化データ（⑥+⑦の横線の領域の符号）が第1のメモリ204へ転送される必要が出てくる。これらを考慮すると、第2のメモリ207だけでなく、第1のメモリ204でも符号化データをファイル或いはパケットとして管理する必要がある。よって、第1のメモリ204にも前述の管理テーブルを有した管理機構が必要となる。

【0084】図16に示された符号化フェーズの状態は、量子化ステップと符号化データの混在の仕方が、再符号化処理の前後で異なっていること以外は、初期状態の符号化フェーズ（図13）と同じである。よって、符号化フェーズ、転送フェーズと符号化・再符号化フェーズを繰り返すことで、最終的に、1ページ分の画像データを設定した上限値以下に圧縮した符号化データを確実に第1のメモリ204に格納することが出来る。

【0085】なお、第1の例の説明とは、転送フェーズと符号化・再符号化フェーズの配置順が逆であることから、図8において転送処理後に行なっていた1ページ分の画像データの入力終了検知（ステップS805）は、符号化・再符号化フェーズで行なう1ページ分の画像データの入力終了検知（ステップS803）と、ほとんど同じタイミングになってしまう。また、2つの検知処理は、機能的にはステップS805と同じで、タイミング的にはステップS803と同じである。従って、これら2つのステップは、新たな1ページ分の画像データの入力終了を検知するステップとして統合し、ステップS1201と表記しておく。

【0086】以上説明した第1、第2の例では、第1のメモリと第2のメモリは物理的に別のメモリであるとして説明をしてきた。これは、2つのメモリに対するアクセスが独立したものとすることができるので有利なためであり、本発明の特徴となす。しかしながら、第1のメモリと第2のメモリを、物理的に別のメモリとしない場合も本発明の範疇に含まれる。物理的に1つのメモリ上に、前記第1のメモリと第2のメモリに相当する2つの領域を確保して、第1のメモリを第1のメモリ領域、第2のメモリを第2のメモリ領域と言い直して、これまでの説明を読み直せば、本発明は、1つのメモリでも実現できることが分かる。

【0087】また、1つのメモリで上記各例を実現する場合には、前記転送フェーズで説明したデータ転送処理のいくつかは不要となる。その詳細はその都度容易に想像できるので説明は省略するが、前記2つの領域を厳密に別けて使用する場合、物理的に2つのメモリを持つ時と同じようにデータ転送処理が必要であるが、2つの領域間で同じデータを共有することになれば、データ転送処理が不要になるだけでなく記憶容量の削減も図れる。

【0088】例えば、第2のメモリ領域で保持していた符号化データを、第1のメモリ領域へ転送する際、該符号化データが格納されている先頭アドレスとデータサイズの2つの情報を第2のメモリ制御部から第1のメモリ制御部へ転送するだけで、前記符号化データを転送したのと同じ効果が得られる。

【0089】前記符号化データを、ファイル形式やパケット形式で格納している場合は、メモリ制御部の間で転送する情報は少し増え、該符号化データに関連する管理テーブル情報を転送する必要がある。それでも、符号化データを転送するよりは、効率が良い。

【0090】さて、上述した画像処理装置によると、入力した画像データを符号化していく際に、目的とするサイズに越えるような場合であっても、その入力を継続しつつ目標とするサイズに収めるよう処理を継続することができるようになるが、本発明では、上記画像データの圧縮データの符号量の制御以外に、該画像データに付随する像域情報を符号化したデータの符号量の制御も行なうものである。

【0091】以下に説明する実施形態では、画像は勿論、その像域情報についても符号化する点に特徴を有する。以下に、その具体的な構成及び動作（もしくは処理）について説明する。

【0092】＜第1の実施形態＞上述の画像処理装置に本発明を適用した第1の実施形態を図17に示す。同図は図1に示した基本構成に本発明を適用したもので、図1の構成と同じ機能ブロックには、同一番号を付すことで、その説明は省略する。

【0093】イメージスキャナやページ記述言語レンダリングなどから入力部101を通して入力した画像データは、既に説明した処理方法に基づいて符号化と再符号化処理を繰り返して行ない、符号化データを設定した符号量以内に収める。

【0094】一方で、画像データは像域情報生成部1701に供給され、前述の像域情報を生成する。本実施形態では、スキャナ入力画像について説明しているので、画像データのみに基づいて像域情報を生成するが、ページ記述言語（PDL）を展開・描画した画像の場合、該PDL情報も参照して像域情報を生成すれば良い。なお、像域情報はスキャナ等の画像入力機器で生成することもある。その場合、前記像域情報も入力部101を通して入力され、像域情報生成部1701を素通りして、次のユニットへ送られる。

【0095】さて、像域情報生成部1701で生成する像域情報であるが、実施形態では、着目要素が文字・線画領域であるか中間調領域であるか、及び、有彩色か無彩色なのかを示す情報を生成する。それぞれ1ビットで表現できるので、1画面について合計2ビットの像域情報（それぞれのビットは、像域情報を構成することになるので像域成分情報という）を生成することになる。

【0096】像域情報生成部1701の動作を簡単に説明すると、次の通りである。

【0097】先ず、文字・線画領域か中間調領域であるのかの判断であるが、文字・線画の場合には背景に対してその輝度(或いは濃度)が急峻に変化する。一方、中間調領域にある場合、注目画素の輝度(或いは濃度)は隣接する画素に対して変化が小さい。従って、注目画素の輝度を L_i 、左右の画素の輝度を L_{i-1} 、 L_{i+1} と定義したとき、

$$|L_i - L_{i-1}| > T$$

又は、

$$|L_i - L_{i+1}| > T$$

である場合に、注目画素は文字線画(のエッジ)であると判断できる。尚、ここの $|x|$ は x の絶対値を示すものである。

【0098】なお、濃度が急峻に変化しているか否かを判断する式は上記のものに限らない。例えば、

$$|2L_i - L_{i-1} - L_{i+1}| > T$$

を満たすかどうかで判断してもよいし、一次元方向のみではなく、2次元方向について判断するようにしても良い(但し、2次元で判断する場合には、像域情報生成部1701内には、複数ライン分の画像データを記憶するためのメモリを必要とする)。

【0099】一方、有彩色・無彩色の判断であるが、入力される画像データはスキャナより読み取られたものであるので、R、G、Bのデータ形式になっている。

【0100】無彩色というのは、RGBの各色成分が互いに同じ輝度である場合であるので、

$$R=G=B$$

なる関係を有する場合には無彩色として判断し、それを満たさない場合には有彩色と判断する。ただし、スキャナ装置が有するCCDの精度も加味する必要があるので、実際は、

$$B-\Delta < R < B+\Delta$$

$$R-\Delta < G < R+\Delta$$

$$G-\Delta < B < G+\Delta$$

の全てを満たすとき無彩色と判断し(Δ は適当な小さい数値)、それ以外を有彩色と判断するようにしても良い。

【0101】場合によっては、RGB色空間を例えば輝度、色相、彩度(例えばLab色空間)に変換し、その彩度が所定値以下の場合に無彩色、所定値を超える場合に有彩色と判断しても良いであろう。

【0102】以上の如く、像域情報生成部1701は、入力された画像データから注目画素が文字・線画/中間調領域か、有彩色か無彩色かを示す2ビットの像域情報を生成し、出力する。

【0103】図17の説明に戻る。上記のようにして、生成した像域情報は、ブロック化ユニット1703にて、まとめて符号化するデータのサイズ、例えば $M \times N$

のサイズ(従って $M \times N \times 2$ ビット)へブロック化する。ブロックサイズは、 32×32 とするが、符号化効率を上げるため、それ以上、例えば 64×64 や 128×128 でも構わないし、必ずしも正方サイズである必要もない。それ故、 $M \times N$ サイズとした。

【0104】ところで、像域情報を符号化する場合、通常の画像データの圧縮に利用するJPEGのような多値の非可逆圧縮は適さない。それ故、可逆圧縮であるJBIG、或いはPackBits等のランレングス符号化を用いることとした。可逆符号化部1705では、該当するブロックの像域情報を、この可逆符号化する。

【0105】符号化した像域情報は、第1のメモリ制御部103を経由して、第1のメモリ104に格納すると共に、第2のメモリ制御部105を経由して、第2のメモリ106にも格納する。つまり、符号化した像域情報も、符号化した画像データと同様、2つのメモリに同時に格納されることになる。また、これと同時に、可逆符号化部1705から出力される符号のサイズ情報を第4のカウント1707にて累積カウントし(1ページの読み取りを開始する際にリセットされる)、その結果を可逆符号化制御部1709に供給する。

【0106】可逆符号化制御部1709内のレジスタには予め目標値(外部より適宜変更することも可能)が設定される。そして、前記像域情報の符号量が該目標値をオーバーした時に、第1のメモリ制御部103、第2のメモリ制御部105、そして、可逆符号化部1705及び可逆符号際符号化部1711に制御信号を出力する。なお、このとき、第4のカウント1707をリセットする。

【0107】以下、この制御信号を受けた場合の、第1のメモリ制御部103、第2のメモリ制御部105、可逆符号化部1705及び可逆符号際符号化部1711それぞれの動作を以下に説明する。

【0108】先ず、第1のメモリ制御部103であるが、上記制御信号(目標値オーバーを示す信号)を受け取ると、第1のメモリ104に格納済みの符号化データ(像域情報)を廃棄する。廃棄の仕方は先に説明した画像データの符号化データを廃棄する方法と同じである。

【0109】また、第2のメモリ制御部105が、この制御信号を受けると、第2のメモリ106から符号化した像域情報を読み出し、その情報を可逆符号再符号化部1711に送る処理を行う。

【0110】可逆符号再符号化部1711は、第2のメモリ制御部105より符号化データを受け取るとそれを復号化し、像域情報の一部を廃棄するか固定値に置き換えた後、再び可逆符号化を行う。詳細は後述するが、像域情報の一部を固定値に置き換えた場合でも情報エンタロピーが低下するため、ランレングス符号化後のデータ量は減少する。再符号化後の属性データは、第2のメモリ106に再び格納する(この段階では、第1のメモ

り104には格納しない)。そして、再符号化後の符号量を第5のカウント1713にてカウントする。第5のカウント1713は、目標値オーバーと判断された時点で、第2のメモリに格納されている既に符号化したデータを再符号化が完了するまでカウントし、再符号化が完了したときに、第4のカウント1707に足しこむ。また、この再符号化が完了したとき、第2のメモリ106に格納された、目標値オーバー検出以前に入力された像域情報の再符号化データを、第2のメモリ106から第1のメモリ104に一気に転送する。

【0111】一方、可逆符号化部1705が目標値オーバーを示す信号を受信すると、それ以降に入力される像域情報に対して、可逆符号再符号化部1711と同様、すなわち、入力される像域情報を変換し(エントロピー低下させる処理を行い)、それを符号化する。

【0112】これ以降、再度目標値オーバーしたことが検出すると、可逆符号化制御部1709は、上記と同様の制御信号を発生することを繰り返す。ただし、可逆符号化部1705及び可逆符号再符号化部1711は、それまでもより、より圧縮率が高くなるよう、像域成分情報の一部を変更もしくは変換するようにする。

【0113】以上の動作をより分かりやすく説明すると次の通りである。

【0114】可逆符号化部1705で符号化された像域情報は第1のメモリ104及び第2のメモリ106にそれぞれ格納される。可逆符号化制御部1709は、像域情報の符号量が目標値オーバーを検出すると、まず、それまでの符号化された第1のメモリ104内の像域情報を破棄させると共に、それ以降に入力される像域情報中の像域成分情報の1つを固定値に変換し、効率の良いランレングスとなるように変更した後に符号化するよう可逆符号化部1709に制御信号を発行する。

【0115】この結果、目標値オーバーとなった後に入力される像域情報は、より高い圧縮率となって可逆符号化され、第1のメモリ104に格納されるようになる。第4のカウント1707は、目標値オーバーとなったことが検出されるとリセットされ、カウントを開始するが、以下に説明する第5のカウントの値を足しこむことで、1ページの原稿画像の最初からカウントしたのと同様の情報量をカウントすることになる。

【0116】一方、目標値オーバーとなる以前の像域情報は、第1のメモリ104から破棄されるものの、第2のメモリ106には残っている。そこで、この第2のメモリ106に格納されていた符号化済み像域情報を、可逆符号再符号化部1711は受信して復号し、そして、符号化方法が変更された可逆符号化部1705と同様の符号化を行って第2のメモリ106に格納していく。このとき、可逆符号再符号化部1711で再符号化した符号量は第5のカウント1713でカウントされ、可逆符号再符号化部1711による再符号化処理が完了する

と、第5のカウント1711のカウンタ結果が第4のカウントに足しこまれる。これにより、第4のカウントは、原稿画像を最初から符号化したときと同じ符号量を計数を行うことになる。

【0117】こうして、可逆符号再符号化部1711による、目標値オーバー検出以前に符号化された像域情報の再符号化が完了すると、その符号化された像域情報は第2のメモリ106から第1のメモリに転送(コピー)されると、可逆符号化部1705から出力される符号化された像域情報は、第1のメモリ104、第2のメモリ105の双方に格納されていく。

【0118】この結果、第1のメモリ104には、目標値オーバーとなる以前の像域情報が再符号化されて格納されることになるので、結果的に、第1のメモリ104には、原稿画像の先頭からの現在入力中の像域情報の符号化後のデータが格納されることになる。

【0119】しかし、第1のメモリ104には、像域情報と実質的に同じ処理によって圧縮された原稿画像(実施形態では非可逆符号)も格納されているわけであるから、第1のメモリ104には、圧縮画像データ及び圧縮像域情報の両方が、それぞれの目標値以下の情報量となって格納されることになる。

【0120】さて、可逆符号化部1705及び可逆符号再符号化部1711において、ランレングスを極くため(可逆圧縮でありながら、圧縮率を高める処理)の具体的な手段であるが、実施形態では次の2つを用意した。処理P1: 文字・線画/中間調画素の識別ビットが中間調画素を示している画素位置の、有彩色/無彩色の識別情報を全て有彩色に変更する。

処理P2: 文字・線画/中間調画素の識別ビットを中間調画素を示すように変更する。

【0121】符号化制御部1713は、或るページの像域情報を符号化している最中に、最初に目標値をオーバーすると判断した場合には、上記の処理P1を採用するよう、可逆符号化部1705及び可逆符号再符号化部1711を制御する。そして、この処理を行っても、目標値をオーバーすると判断した場合(2回目の目標値オーバーと判断した場合)には、処理P2を用いて再度符号化するよう、可逆符号化部1705及び可逆符号再符号化部1711を制御する。

【0122】有彩色/無彩色の識別情報を有彩色にするのは、色空間において有彩色空間が無彩色を含んでいるため、大きな問題にはならないからである。また、文字・線画/中間調の識別情報を中間調にすることも、同様の理由による。

【0123】いずれにしても、かかる像域情報を変更することで、そのエントロピーが低下するため、ランレングス符号化後の情報量は段階的に減少する。

【0124】以上の結果、入力部101からの画像データの入力を継続しつつ、その画像データ及び像域情報の

双方を、それぞれ目標値（目標サイズ）以内に符号化させることができるようになる。

【0125】次に、本第1の実施形態の像域情報に対する処理内容を説明する。像域情報の符号化処理は、以下に示す像域情報特有の処理を除いて、図8に示す画像データの符号化処理とはほぼ同じ手順と同じである。

(1) 符号化処理と可逆符号化処理

(2) 量子化ステップ変更と像域情報変換処理の変更

(3) 再符号化処理と可逆符号再符号化処理

上記処理の相違を、前記図8のフローチャートに反映させたものが図18のフローチャートである。図8におけるステップS303、S309、S311、S313の処理を、図18ではステップS1803、S1809、S1811、S1813の処理に置き換えている。符号化パラメータの初期設定では、画像サイズによって決まる可逆符号化データ量の上限値を可逆符号化制御部内のレジスタにセットし、上記像域情報変換処理の内容を初期状態にリセットする。あとは、図8のフローチャートと同じ説明になる。すなわち、像域情報が目標値を越えた場合の、可逆符号化部1705（及び可逆符号再符号化部1711）における像域成分の変換処理は、ステップS1809で行われることになる。

【0126】次に、1画素あたり2ビットの像域情報に6ビットデータ“000000”を付加して8ビット化した後、可逆符号としてPackbits符号化を行なった場合の、本実施形態における具体的な処理内容を、図19を用いてさらに詳しく説明する。

【0127】Packbits符号化する前の8ビットデータは、図19(a)に示すように上位6ビットは全て0で、下位2ビットの上位側（ビット1）に、対応する画素データが文字・線画であるか中間領域であるかを示すフラグ、下位側（ビット0=LSB）には有彩色か無彩色を表わすフラグデータが入っている。よって、該8ビットデータが取り得る値は、0以上3以下の値である。

【0128】像域情報生成部1701からは、上記8ビットのデータが画素単位で出力されるものとする。具体的な出力データとして、図19(b)に示すデータを考える。

【0129】これをPackbits符号化すると図19(c)に示すデータに圧縮される。圧縮後のデータで負の値は連続するデータの個数を表わし、非連続データの個数は正の値で表わしている。これらは長さ情報と言うもので、該長さ情報のサインビットから連続データが続くのか、非連続データが続くのかを判別することが出来るようになっていく。圧縮後の各データは図9(b)と同じく8ビット（1バイト）である。1バイトの長さ情報で、表わすことができる最大値は255の半分の約128であり、長さ情報がそれ以下の場合は1組の長さ情報とそれに続く像域フラグデータ群で符号化でき、そ

れを超える場合は複数の組の、長さ情報+像域フラグデータ群に分けて符号化される。

【0130】図19(c)の圧縮データを詳しく見てみることにする。最初の長さ情報“4”はマイナスの値なので、上述したように連続データの連続個数を表わし、長さ情報直後の像域フラグデータ“1”が4つ続くことを表わしている。

【0131】次のデータ“4”はまた長さ情報であるが、今度はプラスの値なので非連続データが4つ続くことを示している。よって、前記“4”に続く4つのデータ“2, 3, 2, 3”が非連続データを表わす。図19(c)では、長さ情報と像域フラグデータとが区別しやすいように、プラスの長さ情報のみ下線を引いている。

【0132】上記非連続データの次の“5”は又、連続データの長さ情報で、該長さ情報直後の像域フラグデータ“2”が5つ続くことを表わしている。次の下線付きのデータ“3”は非連続データの長さ情報で、後続する3つのデータ“1, 0, 1”が像域フラグデータであり、さらに次の“-6, 0”は、データ“0”が6個連続することを示している。

【0133】上記圧縮データを可逆符号再符号化部1715で再符号化処理するとどのようなものか、図19(d)(e)を用いて説明する。ここでは、再符号化処理において、有彩色・無彩色フラグを“1”に固定してすべて有彩色にする場合について説明する。

【0134】符号化された像域情報は一旦復号され、図19(b)のデータに戻された後、上記フラグデータの置き換えが行なわれ、図19(d)のデータに変換される。そして、変換されたデータを再びPackbits符号化することによって、図19(e)の符号化データが得られる。再符号化前の15バイトの符号化データが再符号化後は6バイトに減少することが解かる。

【0135】上記の再符号化処理を行なったにもかかわらず、全符号量の計数値が符号化制御部1713内のレジスタに設定された目標値を再びオーバーした場合には、前記再符号化処理が終了していれば、直ちに次の新たな再符号化処理を開始する。前記再符号化処理が終了していなければ、該再符号化処理が終了後、直ちに次の新たな再符号化処理を開始する。

【0136】新たな再符号化処理では、残りの1ビットの像域フラグも“1”に置き換える。これにより、すべての像域フラグデータ（8ビット）の値は“3”となり、データのバイト数をNとすると、符号化後のデータ量はおよそ $(2N/128) + 2$ バイトとなる。

【0137】これは、連続データの個数が128個を超えるたびに、あらたな2バイト1組の符号化データ（長さ情報と連続データ）が増えるためである。

【0138】Packbitsの符号化回路や復号回路それにデータ変換回路はそれぞれ公知の技術であるため、個別の回路構成についての説明は省略する。

【0139】上記説明では簡単化のため、各画素の像域フラグを2ビットとして説明したが、前述したように像域フラグとして他の情報もいくつかある。上記再符号化処理では、2ビットの像域フラグデータでは最大2回の再符号化、4ビットの像域フラグデータでは最大4回の再符号化処理が可能であり、像域フラグのビット数が多い程、再符号化処理の回数を増やすことができ、符号量を多段階で制御することができる。

【0140】さらに、状態数を縮退させる方法を用いれば、さらに符号量を多段階で制御することができる。例えば、2ビットの像域フラグでは、4状態を表わすことができるが、これを1回目の再符号化処理で3状態に縮退させ、2回目の再符号化処理で2状態に縮退させることで、符号化前の情報エントロピーを少しずつ減らし、符号化後のデータ量(符号量)を細かく減らしていくものである。

【0141】上述したフラグデータを1ビットずつ固定値に置き換える処理を、状態数という言葉を用いて表現すると、像域フラグデータを再符号化することに状態数を半分に減らす、と言うことができる。

【0142】1回の再符号化処理で、状態数を半分に減らすより、状態数を1つつ減らした方が、符号量を細かく減らせるのは当然である。

【0143】状態数を1つつ減らした場合の処理結果を図20(b)(c)(d)(e)に示し、それについて説明する。図20(a)は、図19(b)に示した像域フラグデータと同じ、2ビットの4状態の全てが存在するデータである。該4状態をあらためて列挙すると、以下のようになる。

- (1) 有彩色の文字部(データ“3”に対応)
- (2) 無彩色の文字部(データ“2”に対応)
- (3) 有彩色の非文字部(データ“1”に対応、有彩色の画像部とも言う)
- (4) 無彩色の非文字部(データ“0”に対応、無彩色の画像部とも言う)

1回目の再符号化処理で、上記4状態の内、(3)有彩色の非文字部と(4)無彩色の非文字部、の2状態を合体して1つの状態(3')非文字部に縮退させる。これにより、以下の3つの状態となる。

- (1) 有彩色の文字部
- (2) 無彩色の文字部、
- (3') 非文字部

具体的には、データ“0”を“1”に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図20(b)に示すデータに変わる。これをPackbits符号化すると図20(c)に示す符号化データになる。再符号化前の符号化データ図19(c)よりも多少符号量が少なくなっていることが解かる。

【0144】2回目の再符号化処理では、(1)有彩色の文字部と(2)無彩色の文字部、の2状態を合体して

1つの状態(1')文字部に縮退させる。これにより、以下の2状態となる。

- 【0145】(1')文字部
- (3')非文字部

今度は、データ“2”を“3”に置き換えることで、上記状態の縮退を行なう。状態縮退後のデータは図20(d)に示すデータに変わる。このデータは図19(d)のデータと同じである。これをPackbits符号化した符号化データ図20(e)は当然図19(e)と同じ符号化データになる。

【0146】状態数を1つつ減らした場合、再符号化前の15バイトのデータが1回目の再符号化で11バイトに減少し、2回目の再符号化でやっと6バイトに減少する。このように細かく変化して減少する符号量を得ることが出来るので、目標符号量に近い像域フラグの圧縮データを得ることができる。

【0147】上述したように像域フラグデータの可逆符号化処理は、画像データの圧縮符号化処理とは独立に制御され、それぞれに目標符号量以内のデータに収められる。

【0148】符号化された2種類のデータは、外部に接続されるネットワーク機器、画像出力装置や大容量記憶装置等へ出力する際に多重化する。該多重化を考慮して、前記2種類のデータを符号化処理する単位を、前述したように同じサイズに合わせておき、1単位を符号化して生成される符号化データを1つのパケットあるいはファイルとして管理・格納する。多重化する際に、画像位置が同じ2種類のパケットデータを、例えば画像データ・像域データの順に連結して、1パケット化し、外部へ出力する。

【0149】<第2の実施形態>第2の実施形態は、図2の構成に対して本発明を適用したもので、その構成を図21に示す。同図には図2及び図17で用いた機能ブロックを使用している。該ブロックには同一番号を付し説明を省略する。

【0150】本第2の実施形態では、図2の構成における画像データの圧縮符号化処理と同じように、2つの符号化部で情報エントロピーの異なる2種類の像域情報を可逆符号化する。この2つの符号化部とは第1の可逆符号化部1705と第2の可逆符号化部2106である。本第2の実施形態で追加する第2の可逆変換部2106では、第1の可逆符号化部よりもエントロピーの低い像域情報を符号化する。

【0151】本実施形態において追加される、その他のブロックは、上記第2の可逆変換部2106が出力する符号量をカウントする第6のカウント2108のみである。

【0152】図21において、画像データを圧縮符号化するブロックと像域情報を可逆符号化するブロックとは以下のような対応関係がある。

(1) 第1、第2の符号化部と第1、第2の可逆符号化部

(2) 第1、第2、第3のカウントと第4、第6、第5のカウント

(3) 再符号化部と可逆符号再符号化部

(4) 符号化シーケンス制御部と可逆符号化制御部

上記各項目の前者のブロックを用いて画像データを圧縮符号化し、後者のブロックを用いて像域情報を可逆符号化する。2つの符号化処理は、ほとんど同じ方法で独立に制御され、各々のデータを目標符号量以下に圧縮する。

【0153】画像データは量子化ステップの変更により段階的に符号量を減らしていくが、像域情報の方は、可逆符号化前の像域情報のフラグデータの状態数を減少させることで、可逆符号化後の符号量を減らしていく。

【0154】初期状態では、第1の可逆符号化部1705は全像域情報のフラグデータの状態数を前記第1の実施形態で述べた像域情報交換処理によって減らし、それから可逆符号化を行なう。

【0155】第1の可逆符号化部から出力した符号量が目標値をオーバーしたところで、第1のメモリ104に格納していた可逆符号を廃棄して、第2のメモリ106に格納していた可逆符号を第1のメモリに転送し、前記第2の可逆符号化部で行なっていた像域情報交換処理を第1の可逆符号化で引き継いで、前記第1のメモリに転送した可逆符号の後に符号化データを格納する。

【0156】第2の可逆符号化部では、これまでよりもさらに前記フラグデータの状態数を減らすような像域情報交換処理を行ない、それを可逆符号化する。そうすることで、第1の可逆符号化部より、符号量が少ない可逆符号を生成して第2のメモリに格納する。

【0157】すでに第2の可逆符号化部で符号化済みの第2のメモリ内の可逆符号、すなわち第1のメモリにも転送した可逆符号は、可逆符号再符号化部1711にて復号化し、第2の可逆符号化部で行なう像域情報交換処理と同じ状態数となるように可逆符号再符号化部にて像域情報交換処理し、再び可逆符号化して、第2のメモリに書き戻す。このように、本実施形態においても、画像データの圧縮符号化処理と同様の方法で、像域情報の符号化処理を行ない、該符号量を制御することが出来る。

【0158】よって、上述した処理のフローチャートは、図12に示す画像データを圧縮符号化処理するフローチャートの一部の表現を変えるだけで、そのまま用いることが出来る。変える部分は第1の実施形態の場合と同様、以下の3点である。

(1) 符号化処理 → 可逆符号化処理

(2) 量子化ステップ変更 → 像域情報交換処理の変更

(3) 再符号化処理 → 可逆符号再符号化処理

上記3点の表現を変えたフローチャートを図22に示す。ステップS2103、S2109、S2111、S2113の4つの処理内容が上記のように変わる。

【0159】画像データの圧縮符号化処理と像域情報の符号化処理は並列に行ない、且つ独立に制御するので、図21に示す本第2の実施形態は、図12と図22の2つのフローチャートによって処理フローを規定することが出来る。

【0160】以上説明した第1、第2の実施形態の処理は、始めに説明した基本部分を含めて、マルチタスクOSを搭載した汎用の情報処理装置（例えばパーソナルコンピュータ等）によるコンピュータプログラムによっても実現できる。従って、本発明はかかるコンピュータプログラムにも適用できるものである。この場合、図1や、第1、第2の実施形態で示したブロック構成図における各ユニットは、コンピュータプログラムのモジュールもしくは関数プログラムで実現できるのは、当業者であれば容易に理解できよう。従って、本願発明は、コンピュータプログラムに適用しても構わない。

【0161】また、通常、パーソナルコンピュータ等の汎用情報処理装置にコンピュータプログラムを導入する場合には、そのコンピュータプログラムを記憶する記憶媒体（フロッピー（登録商標）ディスク、CDROM、MO等）をセットし、インストール或いはコピーすることで実現できるものであるから、本発明はかかる記憶媒体をもその範疇とするものである。

【0162】以上説明したように本実施形態によれば、可逆符号化した像域情報を復号する復号化手段と、像域情報を該像域情報の情報エントロピーが少なくなるように該像域情報の一部を書き換え、或いは削除する像域情報交換手段と、前記復号化手段で復号した像域情報を前記像域情報交換手段で交換した該像域情報を再び可逆符号化する再符号化手段と、前記情報交換手段を備えた可逆符号化手段と、少なくとも1ページ分の画像データに付随する像域情報を符号化したデータを格納することが可能な格納手段とを有し、前記符号化データの量に応じて、前記可逆交換手段が備えた像域情報交換手段と、再符号化手段の前段にある像域情報交換手段の双方を制御することにより、1ページ分の像域情報を所望の符号化データ量に収めることができる。

【0163】

【発明の効果】以上説明したように本発明によれば、多値画像データを入力しなおすことなく、多値画像データ及びその像域情報それぞれを目的とするサイズ内に収めるよう符号化することが可能となる。

【図面の簡単な説明】

【図1】本発明を適用する画像処理装置の第1の基本構成を示す図である。

【図2】本発明を適用する画像処理装置の第2の基本構成を示す図である。

【図3】図1の構成における処理を簡略化して示したフローチャートである。

【図4】初期状態の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図5】符号化・再符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図6】転送フェーズにおけるデータフローとメモリ内容を表わす図である。

【図7】転送フェーズ後の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図8】図1の構成における処理の詳細を示すフローチャートである。

【図9】図1の構成の変形例における符号化・再符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図10】図9の変形例における転送フェーズにおけるデータフローとメモリ内容を表わす図である。

【図11】図9の変形例における転送フェーズ後の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図12】図2の構成における処理手順を示すフローチャートである。

【図13】図2の構成における、初期状態の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

ーズにおけるデータフローとメモリ内容を表わす図である。

【図14】図2に構成における、転送フェーズにおけるデータフローとメモリ内容を表わす図である。

【図15】図2の構成における、符号化・再符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図16】図2の構成における、符号化・再符号化フェーズ後の符号化フェーズにおけるデータフローとメモリ内容を表わす図である。

【図17】本発明の第1の実施形態における装置のブロック構成図である。

【図18】第1の実施形態における処理手順を示すフローチャートである。

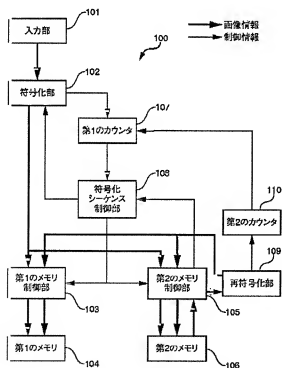
【図19】第1の実施形態における可逆符号の符号化データと再符号化後の符号化データを示す図である。

【図20】可逆符号の再符号化後の符号化データと再々符号化後の符号化データの他の例を示す図である。

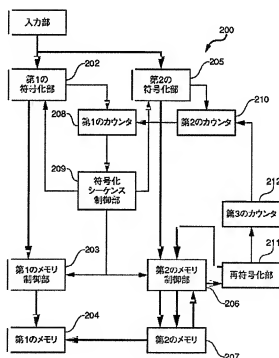
【図21】第2の実施形態における可逆符号の再符号化後の符号化データと再々符号化後の符号化データを表わす図である。

【図22】第2の実施形態における処理手順を示すフローチャートである。

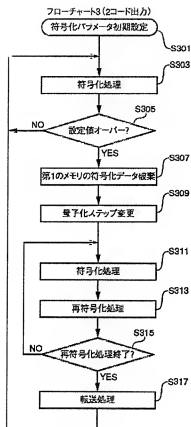
【図1】



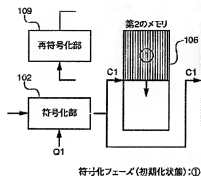
【図2】



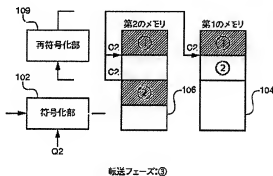
【図3】



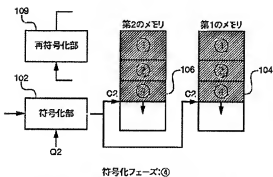
【図4】



【図6】

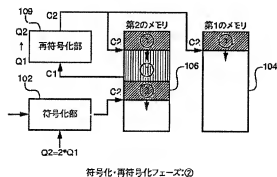


【図7】



※メモリ内の番号①②③④は該処理段階で符号化された時のフェーズを表し、再符号化された時のフェーズ番号は反映していない

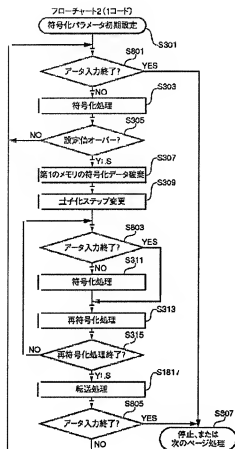
【図5】



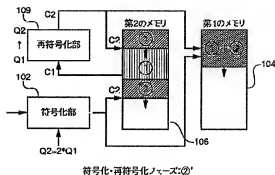
【図20】

- (a) 1, 1, 1, 1, 2, 3, 2, 2, 2, 2, 2, 2, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2,
- (b) 1, 1, 1, 1, 2, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2,
- (c) -4, 1, 4, 2, 3, 2, 3, -5, 2, -9, 0,
- (d) 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3,
- (e) -4, 1, -9, 3, -9, 1,

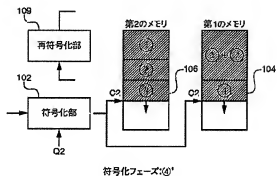
【図8】



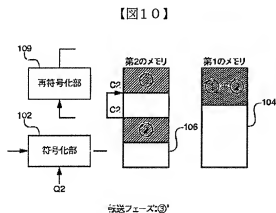
【図9】



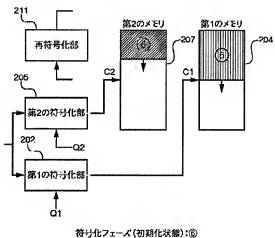
【図11】



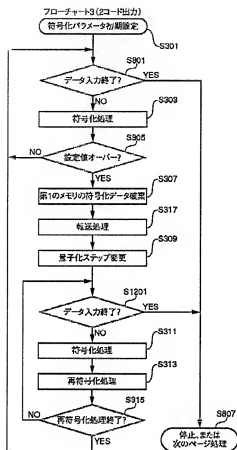
【図13】



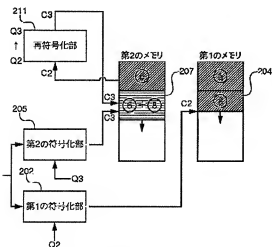
符号化フェーズ(初期化状態):②



【図12】

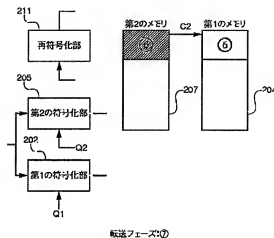


【図15】



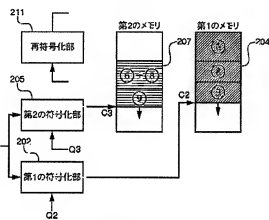
符号化-再符号化フェーズ③

【図14】



転送フェーズ②

【図16】

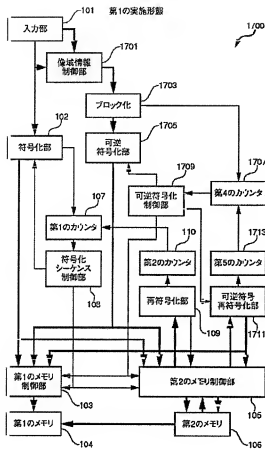


符号化フェーズ③

【図19】

- Legend for Figure 19:
- 1/0: 文字紙/表文字旗 (Text Paper/Text Flag)
 - 右彩色/無彩色フラグ (Right Color/No Color Flag)
- (a) 000000BB (8ビット)
- (b) 1, 1, 1, 1, 2, 3, 2, 2, 2, 2, 2, 1, 0, 1, 0, 0, 0, 0, 0, 2
- (c) -4, 1, 4, 2, 3, 2, 3, -5, 2, 3, 1, 0, 1, -6, 0
- (d) 1, 1, 1, 1, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 3
- (e) -4, 1, -9, 3, -9, 1

【図17】



【図18】

